

PRÁCTICA 2: PUENTE DE ÁMBITE



- un coche en una dirección puede cruzar el puente cuando no viene ningún coche en dirección contraria (solo un coche cruzando el puente)
- un peatón puede cruzar en cualquier dirección cuando no haya ningún coche cruzando y pueden cruzar 2 peatones a la vez en cualquier dirección

SOLUCIÓN sencilla: por turnos van cruzando coches y peatones, uno a uno en la dirección que elijan

el n° de coches y peatones que quieren cruzar al principio siempre será ≥ 0
INV: $\langle \text{cargonorte} \geq 0, \text{cargosur} \geq 0, \text{peaton} \geq 0, \text{ (inicialización)}$
 $\text{cruzandonorte} > 0 \rightarrow \text{cruzandosur} = 0 \wedge \text{cruzandopeaton} = 0,$
 $\text{cruzandosur} > 0 \rightarrow \text{cruzandonorte} = 0 \wedge \text{cruzandopeaton} = 0,$
 $\text{cruzandopeaton} > 0 \rightarrow \text{cruzandonorte} = 0 \wedge \text{cruzandosur} = 0 \quad \vee$

esta solución cumple la seguridad ya que no cruza más de un coche a la vez (si hay algo cruzando, el resto están esperando) y si cruzan peatones no hay coches cruzando.

DEADLOCK

- solo ocurre un proceso (cruce) como máximo en cada momento (exclusión mutua) de coches y peatones
- cada proceso utiliza sus propios recursos
- existe una espera circular: los procesos (cruces) bloqueados o en espera retienen los recursos que tienen asignados hasta que hayan terminado el proceso (ya cruzado) y se los pasen al siguiente proceso

INANICIÓN: habiendo definido los "turnos" o método de cruce, evitamos el problema de inanición (que todos los elems quieran cruzar a la vez). En este caso, hemos definido los cruces de 1 en 1 o otra alternativa tal que se pueda establecer un orden y prioridad.